# Models of Computation made easy

Mike Spivey,

Michaelmas Term, 2014

Revised Michaelmas Term, 2021

There are a couple of places where arguments given in the notes for *Models of Computation* could be streamlined. I'm recording the suggestions here as a help to my own pupils and a suggestion for teaching the course in the future.

## 1 Product construction

A product construction proves using DFA's that the class of regular languages is closed under union. If $M_1 = (Q^1, \Sigma, \delta^1, q_0^1, F^1)$ and $M_2 = (Q^2, \Sigma, \delta^2, q_0^2, F^2)$ are two DFA's then we can construct a DFA $M$ that accepts $L(M_1) \cup L(M_2)$ by setting $M = (Q, \Sigma, \delta, q_0, F)$, where $Q = Q^1 \times Q^2$ and

$$\delta(\langle q_1, q_2\rangle, x) = \langle \delta^1(q_1, x), \delta^2(q_2, x)\rangle)$$

and $q_0 = \langle q_0^1, q_0^2\rangle$ and

$$F = (F^1 \times Q^2) \cup (Q^1 \times F^2) = \{\, \langle q_1, q_2\rangle \in Q \mid q_1 \in F^1 \text{ or } q_2 \in F^2 \,\}.$$

The notes give a lengthy proof that $L(M) = L(M_1) \cup L(M_2)$ by talking about sequences of states and showing inclusion both ways, when in fact there is a much shorter proof. For observe that $w \in L(M_1)$ if and only if $\hat{\delta}^1(q_0^1, w) \in F^1$, where $\hat{\delta}^1$ is the obvious extension of $\delta^1$ to strings $w$; the same holds for $M_2$ and for $M$. It's easy to see, because the $\delta$ for $M$ acts independently on the two components, that

$$\hat{\delta}(\langle q_1, q_2\rangle, w) = \langle \hat{\delta}^1(q_1, w), \hat{\delta}^2(q_2, w)\rangle.$$

Thus we may reason as follows:

$$\begin{aligned}
w \in L(M) &\iff \hat{\delta}(q_0, w) \in F \\
&\iff \langle \hat{\delta}^1(q_0^1, w), \hat{\delta}^2(q_0^2, w)\rangle \in (F^1 \times Q^2) \cup (Q^1 \times F^2) \\
&\iff \hat{\delta}^1(q_0^1, w) \in F^1 \text{ or } \hat{\delta}^2(q_0^2, w) \in F^2 \\
&\iff w \in L(M_1) \text{ or } w \in L(M_2).
\end{aligned}$$

No explicit sequences of states, no dot-dot-dot's are needed, just a simple chain of equivalences.

## 2 Pumping lemma

As stated in the course, the pumping lemma is as follows: Let $L$ be any regular language; then $L$ has a pumping length $p$ such that

> If $w \in L$ and $|w| \geq p$,
> then $w$ can be written $w = xyz$ with $y \neq \epsilon$ and $|xy| \leq p$,
> so that $xy^kz \in L$ for all $k \geq 0$.

This formulation is valid, but attempting to apply it in concrete situations leads to knavish tricks. For example, to show that the language $L_1 = \{\, 0^m 1^n \mid m \neq n \,\}$ is not regular, one is led to consider the string $0^{p!}1^{2p!}$, with the factorials appearing simply in order to enable the two blocks to match in length after pumping.[1] This knavishness can be avoided if we use a corollary to the pumping lemma: every regular language $L$ has a pumping length $p$ such that:

> If $u$ is a string with $|u| \geq p$,
> then $u$ can be written $u = xyz$ with $y \neq \epsilon$,
> so that if $uv \in L$ for any string $v$,
> then $xy^kzv \in L$ for all $k \geq 0$.

The proof is essentially the same, but the applications are easier. In the case of language $L_1$, we begin with $u = 0^p$ so that $x = 0^a$, $y = 0^b$, $z = 0^c$ with $p = a + b + c$ and $b \neq 0$. Consider the string $v = 1^q$ where $q = p + b$. We have $uv = 0^p1^q \in L_1$, so also $xy^2zv = 0^q1^q \in L_1$, a contradiction. No factorials required.

For the proof, let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA that accepts $L$, let $p = |Q|$, and consider any string $u \in \Sigma^*$ with $|u| = n \geq p$. When reading $u$, the DFA begins in state $q_0$ and enters states $q_1 = \delta(q_0, u_0)$, $q_2 = \delta(q_1, u_1)$, $\ldots$, $q_n = \delta(q_{n-1}, u_{n-1})$. Since $n \geq p$, this sequence contains repeated states, say $q_i = q_j$ with $0 \leq i < j \leq n$. Let $x = u_{[0..i)}$, $y = u_{[i..j)}$, $z = u_{[j..n)}$, so that $u = xyz$ and $y \neq \epsilon$. If $v$ is such that $uv \in L$ then $q_n \xrightarrow{v} q' \in F$ and so

$$q_0 \xrightarrow{x} \underbrace{q_i \xrightarrow{y} q_j = q_i \xrightarrow{y} \ldots \xrightarrow{y} q_j}_{k \text{ steps}} \xrightarrow{z} q_n \xrightarrow{v} q' \in F,$$

and $xy^kzv \in L$ as required.

---

[1] I call this trick knavish because, although it is ingenious and works for this example, other examples that are as obviously non-regular will require an inexhaustible repertoire of other tricks.